MULTIMEDIA UNIVERSITY

TABLE NO: | | |

STUDENT ID NO: | | | | | | | | | | |

VENUE: _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2015/2016

TCP1101 – PROGRAMMING FUNDAMENTALS
TCP1231 – COMPUTER PROGRAMMING I

(All sections / Groups)

1st Mac 2016
2.30 pm – 4.30 pm
(2 Hours)

Question	Mark
1	
2	
3	
4	

Examiner 1 Signature: _____

Examiner 2 Signature: _____

Total: _____

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 12 printed pages (including cover page) with 4 Questions only.

2. Student must attempt **ALL QUESTIONS**. All questions carry equal marks (10 marks). The distribution of the marks for each question is given.

3. Please print all your answers **CLEARLY** in the specific answer boxes provided for each question. Submit this question paper at the end of the examination.

Unless otherwise stated, every code segment provided in every question is contained within a main function, complete with the directive `#include <iostream>` and the statements `using namespace std;` and `return 0;`

QUESTION 1

(a) Write the output as the result of executing the following program. [2]

```
#include <iostream>
using namespace std;

int main(){
   int x = 3, y = 0, z = 6;

   if (x > y)
        if (x > z)
             cout << x << endl;
        cout << y << endl;
   cout << z << endl;

   {
        int x = 5;
        cout << x << endl;
   }
   cout << x << endl;

   return 0;
}
```

(b) Rewrite the following *do-while-loop* by using a *for-loop*. [2]

```
#include <iostream>
using namespace std;
int main(){
   int records = 500 ;
   do
   {
        cout << records << " records processed.\n" ;
        records-- ;
```

```
    } while (records > 0);
    return 0;
}
```

(c) Write a C++ code segment by using loop structures to produce the output pattern given below. The user should be able to set the number of lines (rows). In the given sample output pattern, the user sets the number of lines to four. [3]

```
#
###
#####
#######
```

(d) Draw a flowchart to accept a series of positive integers from the user, and then output the average of all the numbers entered by the user. User input terminates when any negative number is entered. [3]

Draw your flowchart below:

QUESTION 2

(a) Give **two** reasons why Call-by-Reference in functions can help address the drawbacks of using Call-by-Value. [2]

(b) Write a switch-case block in the program below, to enable it to produce the expected outputs given.

Expected Output 1 (user input in bold):
Enter selection: **n** Normal booking
Expected Output 2 (user input in bold):
Enter selection: **s** Special booking

```
#include <iostream>
using namespace std;

int main()
{
   char sel;
   cout << "Enter selection: ";
   cin >> sel;
```

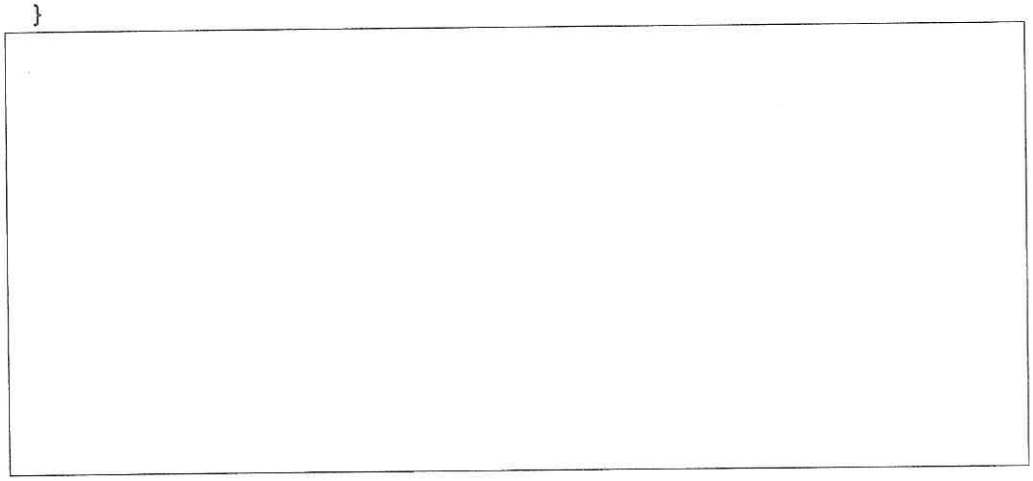
[3]

```
    return 0;
}
```

(c) Complete the following program in the 2 separate sections given. You must overload the function **area** to calculate the area of a circle. You may assume π as 3.14. The expected output and inputs are given after the program. [3]

```
#include <iostream>
using namespace std;

double area(double w, double l) {
   return w * l;
}
```

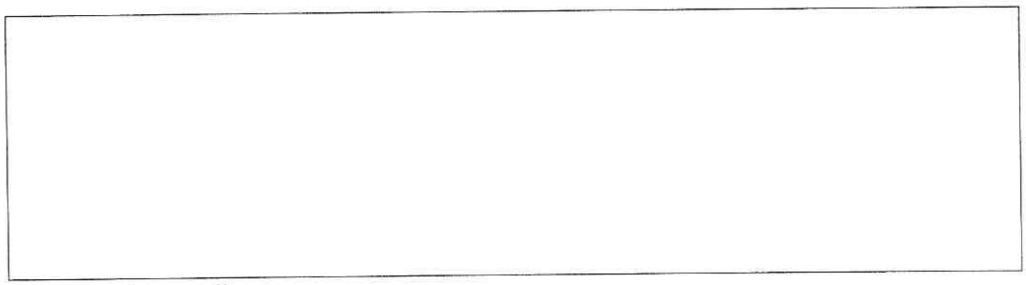


```
int main(){

   double radius, width, length;

   cout << "Enter radius, width and length: ";
   cin >> radius >> width >> length;

   cout << "Area of a rectangle is " << area(width,
   length) <<
            endl;
```



```
    return 0;
}
```

Expected Output/Inputs:

```
Enter radius, width and length: 8 5 9
Area of rectangle is 45
Area of circle is 200.96
```

(d) In the following program, write the output as according to the given user inputs.

```
#include <iostream>
using namespace std;

int main(){
   char key;
   int k = 12;
   cin >> key;

   switch(key) {
        case '#':
             cout << k*2 << endl;
             break;
        case '@':
             cout << k++ << endl;
             break;
        case '*':
        case '%':
             cout << (k+4)/4 << endl;
        default:
             cout << "Do nothing" << endl;
   }
   return 0;
}
```

[2]

Input for *key*: @
Output:

Input for *key*: *
Output:

QUESTION 3

(a) Write the output of the following program: [1]

```
#include <iostream>
using namespace std;

int main()
{
     int i, b = 0;
     int c[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};

     for (i = 0; i < 10; i++)
          if (c[i] % 4 == 0)
               b += c[i];

     cout << b;
     return 0;
}
```

(b) Write a function named **sum** that has the following arguments:

* an array of floating point values;
* an integer that tells how many floating point values are in the array.

The function should return the sum of all the floating point values stored in the array. Thus, for example, if the array that's passed to the function looks like this:

0	1	2	3	4
5.8	2.6	9.0	3.4	7.1

then the function should return the value 27.9 as its value. [2]

(c) Write one advantage of vector over array. [1]

(d) If `a` is an array, and `p` and `q` are pointers pointing to `a`, for each of the following expressions, state whether it is syntactically correct or not (write "Yes" or "No"). [2]

```
i)   a[0] = 5;
ii)  &p = &a[0];
iii) q = &(p + 2);
iv)  *p = *p + 2;
```

i) ii) iii) iv)

(e) Write two versions of a swap function that swaps two integer numbers below. [4]

i. `swap1` using the passed-by-reference mechanism.

ii. `swap2` using the passed-by- address (pointer) mechanism.

QUESTION 4

(a) Consider the following class declaration:

```
class Rectangle
{
public:
      Rectangle();
      void set(double, double);
      double calcPerimeter();
      double calcArea();
private:
      double height;
      double width;
}
```

i. Write a definition of the default constructor which sets both the private member data to 1.0. [2]

ii. Write a definition of the member function `set` which sets both the private attributes according to the input parameters. [2]

iii. Write a definition of the member function `calcArea` which calculates and returns the area of the rectangle object. [2]

iv. Write a main function that instantiates an object `table` of the class `Rectangle`, set the data members height to 3.2 and width to 5.5 using the `set` member function, and then displays the area of the object returned by the member function `calcArea`. [2]

(b) Within a C++ program, create a pointer to an integer, use that pointer to create a dynamic variable of the type int. Assign the value 88 to this dynamic variable. Lastly, destroy that dynamic variable and free up the memory occupied by it. [2]